

GRANT
1N-02-CR
43803

P-77

LV Software Support for Supersonic Flow Analysis

Final Technical Report
October 1990 to October 1991

for
NASA Grant NAG3-1215
Mr. Royce D. Moore, Technical Officer
Turbomachinery Branch, Mail Stop 77/6
NASA Lewis Research Center
21000 Brookpark Road
Cleveland, OH 44135

by
Dr. William A. Bell, Project Director
Georgia Tech Research Institute
Signature Technology Laboratory
Atlanta, GA 30332

Contracting through
Georgia Tech Research Corporation
Centennial Research Building
Georgia Institute of Technology
Atlanta, GA 30332

(NASA-CR-188914) LV SOFTWARE SUPPORT FOR
SUPERSONIC FLOW ANALYSIS Final Technical
Report, Oct. 1990 - Oct. 1991 (Georgia
Inst. of Tech.) 77 p CSCL 01A

N92-10978

Unclas
G3/02 0043803

1.0 Introduction - The Need for an LV System and the Software to Control It

The NASA Lewis Research Center (LeRC) has maintained a leadership position in research into advanced aerospace propulsion systems. For the next generation of aircraft, engine designs continue to involve complex, high-speed flows. Performing the detailed flow diagnostics to properly evaluate these designs requires advanced instrumentation applicable to high-speed, unsteady flows. A laser velocimeter (LV) system is such an instrument. Since the LV is a proven technique providing nonintrusive measurements, it is capable of acquiring the velocity field with minimal interference to the flow.

However, the next generation of aircraft design pushes the limits of present analytical, numerical, and experimental techniques for fluid dynamics. The LV is no exception. To ensure the proper operation of the instrument in diagnosing advanced propulsion systems, the system must be configured, calibrated, and tested in order to ensure accurate results under flow conditions that are near the limits of operation. The measured velocity data must be rapidly acquired and presented to the operator online so that he can respond to changing test conditions. The data must be further processed offline to perform more advanced flow diagnostics such as time series analysis, velocity bias correction, and computation of higher order statistics.

Based on anticipated flow conditions, a laser velocimeter system has been procured from TSI, Inc. and utilizes counter processor technology. To meet the needs of advanced research into propulsion, this instrument must be integrated into an existing VAX/VMS computer system for data acquisition, processing, and presentation.

This report describes the work performed under this NASA Grant, NAG3-1215. During 1991 the software developed allows the operator to configure and checkout the TSI LV system prior to a run. This setup procedure establishes the operating conditions for the TSI MI-990 multichannel interface, and the RMR-1989 rotating machinery resolver. In addition to initializing the instruments, the software package provides a means of specifying LV calibration constants, controlling the sampling process, and identifying the test parameters. In addition, using the extensive computing capabilities at Georgia Tech, a network link established using the TELNET protocol allows transfer of the software to NASA LeRC computing facilities for test and evaluation. In addition, TELNET provides the means for logging into the onsite computers at NASA LeRC from remote locations such as Georgia Tech. Thus, the software can be developed and tested using the equipment at the test site, which minimizes turnaround time and increases the responsiveness to NASA's ongoing research needs.

With the basis now established for controlling the operation of the LV system, software development during 1992 can concentrate on the analysis and presentation of the LV data. Because of the diversity of the types of flows to be investigated, the data analysis includes velocity statistics, time series analysis, and conditional sampling. In addition, the LV is part of an overall instrumentation system for acquiring and analyzing complex, unsteady flow fields of propulsion systems. Consequently, the software design must accommodate the total environment and address integration and coordination issues.

The remainder of this final technical report describes the objectives, approach and progress made under NASA Grant NAG3-1215 during 1991. Section 2 presents the overall objectives as well as the specific objectives for 1991. Section 3 describes the approach taken in the LV software development and Section 4 discusses the progress.

2.0 Objectives for 1991

The primary objectives of this or any other software package for aerodynamic testing are to acquire, analyze, and present experimental data in a form that makes sense to the operator. The primary goals achieved for 1991 are

1. complete the definition of the final computer system requirements;
2. develop software for configuring a TSI LV system prior to a run;
3. provide software for controlling a TSI rotating machinery resolver; and
4. implement the networking procedures for transferring and testing the software to NASA LeRC computer facilities.

3.0 Approach to LV Software Development

The approach to software development for the LV system follows the principles developed in References 1 and 2. These principles support a modular, structured software development process. Since the LV is part of an overall instrumentation suite, these principles facilitate the integration of the LV software into the overall test support software and are discussed in further detail in the programmer's manual included in Appendix 6.1.

As pointed out in Appendices 6.1 and 6.2, software in support of on-line testing consists of three main functions: acquisition, analysis, and presentation. The acquisition process consists of identifying the test, configuring the instrument and initiating data transfers. During 1991, the software development concentrated in this area. The software saves the current test and instrumentation configurations for retrieval, saving the operator from having to re-enter the information after exiting the program. Data analysis and presentation depend on the needs of the end user. For instrument diagnosis and testing, the data analysis and presentation consists of printing out the raw instrument readings for visual inspection. For testing of aeropropulsive systems, the data analysis involves converting the instrument readings to physical quantities, such as velocity and time between samples for the LV, and presenting summary statistics for on-line test monitoring and control.

Because of the nature of the testing environment, it is important that the software be robust. In the present context, robustness means that, at a minimum, the software responds to errors induced by the instrument or operator by issuing an error message to the monitor and returning to a known condition. Using error handling routines that are part of the VAX/VMS operating system, the software attempts to identify the error source, type, and correction. Whenever possible, error recovery is attempted.

As described in the programmer's manual in Appendix 6.1 and the user's manual in Appendix 6.2, the LV software identifies the test and initializes the TSI Model MI-990 multichannel interface and the RMR-1989 rotating machinery resolver. The software followed the procedures recommended in the instrumentation manuals, which are listed in Section 2 of each of the appendices.

3.1 Requirements Dictated by the Operating Environment

Prior to software development, the software and hardware requirements definition led to the use of the existing VAX computers manufactured by the Digital Equipment Corporation. The VMS operating system supports the real-time environment of engine and inlet tests by sophisticated system services for control of data transfers and error detection and recovery. Following the procedures implemented at NASA-LeRC, low-speed data transfers required for instrumentation setup employ the RS-232 serial interface standard. The high-speed data transfers from the counter processor take place across the LVABI (LV Autocovariance Buffer Interface), which uses direct memory access via the DR11-W parallel interface on the VAX. NASA has developed software for the LVABI, which can be integrated into the LV software package developed under this grant as required.

3.2 Design of the LV Data Acquisition Software

The modular design of the LV software allows the components to operate separately for independent configuration and testing or concurrently during a run. This design also facilitates integration with other software packages and instrumentation systems as the need arises.

The main goals of the design are user friendliness and robustness. The VAX/VMS screen management utility supports a menu-driven format, where the available commands are defined on the screen for the operator's convenience. Also, default values are provided where applicable to prevent unbounded entries from inhibiting operation. The screen manager also provides output on terminals compatible with the VT100, which is widely used in emulation software. The input and output routines to the operator's console are independent of the other LV software, so that they can be readily replaced with other programs for screen management as the need arises.

3.3 Code Development

After completion of the design, coding normally occurs in the software development cycle. The VAX version of FORTRAN V with extensions is the programming language used in the LV software. With calls to system services for interrupt and error handling, this programming language handles the interactions between the LV system, the VAX, and the operator. Details of the design philosophy of the code are given in the programmer's manual in Appendix 6.1.

3.4 Testing of the LV Software

Software testing consists of three parts - initial, simulation, and operational. Initial testing consists primarily of finding and correcting errors in the coding or linking of the software. The next level of testing is simulation, where the data transfers involve a local device, such as a terminal, that simulates the operation of the instrument under test. The commands sent to and the response from the device can be tested using this procedure. The performance of the software under both standard and anomalous conditions can be ascertained. Finally, operational testing is performed using the actual instruments under standard operating conditions. Upon completion of this testing, the software is ready for use.

4.0 Progress during 1991

A formal software development cycle consists of four parts - requirements definition, design, coding, and testing. During the requirements definition phase of the LV software development, the final hardware and software systems were selected and procured. The hardware consists of a TSI MI-990 multichannel interface and a RMR-1989 rotating machinery resolver interfaced to a VAX series of computers. Since the VAX family maintains compatibility across product lines, software written on one VAX model transports to another without alteration. This greatly facilitates software development. After reviewing the manuals for the MI-990 and RMR-1989, the commands and techniques for controlling the instrument were established and the software requirements defined using structured specification.²

Following the coding of the routines for identifying and controlling the test, the MI-990 multichannel interface configuration software was coded and verified using a terminal to simulate the response of the instrument to operator commands. The user's manual in Appendix 6.2 show the procedures for configuring the instrument. In addition to fringe crossing times for determining velocity, the multichannel interface provides the time between samples required for the removal of velocity bias³ and time series analysis⁴ of the unsteady velocity data. The coincidence between channels can be set to support the measurement of cross correlations of the velocity data between channels. The software for performing statistical and time series analysis of the velocity data is available and can be integrated into the LV data analysis and presentation software as required.

The RMR-1989 rotating machinery resolver allows the velocity sampling by the LV to be synchronized with the rotation of a turbine shaft. This synchronization allows conditional sampling of the velocity, which has a variety of applications in studies of aeropropulsion and aeroacoustics of engines.⁵ Conditional sampling is a basic technique for interblade flow mapping with the LV. The RMR-1989 provides the capability of setting up the sampling procedure using the commands presented in Appendix 6.2. The conditional sampling techniques and software modules have been developed and can be integrated into the LV software once final data analysis and presentation requirements are established.

After coding of the MI-900 and RMR-1989 configuration software, the individual routines were tested using an external terminal to simulate the operation of the devices. In this

fashion the software could be tested under a wide variety of conditions including its ability to handle erroneous inputs from the simulated device and timing issues. To increase robustness, the software checks for errors in the operator entries to minimize disruption from erroneous inputs.

In order to conduct final, operational testing at the NASA LeRC computing facilities, the software must be transported from the development site at Georgia Tech to the target facilities. An interface to the NASA LeRC computing network was established using TELNET. Not only does this arrangement allow transfer of the software between the development and application facilities, but it also provides a means for operating the software from a remote site. This capability was applied to transfer, compile, and test the LV software on the microVAX at the test facility. The final operational testing of the software will take place upon completion of the final installation and checkout of the LV hardware.

5.0 References

1. Booch, G., *Software Engineering with Ada*, 2nd Edition, Benjamin/Cummings Publishing Company, 1987
2. deMarco, T., *Structured Analysis and System Specification*, Yourdon, Inc., 1979
3. Edwards, R. V. and Meyers, J. F., "An Overview of Particle Sampling Bias", *Second International Symposium on Applications of Laser Anemometry to Fluid Mechanics*, Lisbon, Portugal, July 1984.
4. Bell, W. A., "Spectral Analysis of Laser Velocimeter Data with the Slotted Correlation Method", Paper AIAA-86-1102, AIAA/ASME 4th Fluid Mechanics, Plasma Dynamics, and Lasers Conference, Atlanta, Georgia, May 1986.
5. Lepicovsky, J., Bell, W. A. and Ahuja, K. K., "Conditional Sampling with a Laser Velocimeter and Its Application for Large-Scale Turbulent Structure Measurement". LG83ER0001, Lockheed-Georgia Company, January 1983.

6.0 Appendices

The appendices consist of two documents. The first is the programmer's manual for the LV software, and the second consists of the user's manual. These manuals describe the existing capability of the LV software and points out routines that can be included to meet future requirements.

6.1 Programmer's Manual

The programmer's manual describes the routines that make up the LV software developed during 1991 under NASA Grant NASG3-1215.

1.0	SCOPE	1
1.1	IDENTIFICATION	1
1.2	PURPOSE	1
1.3	BACKGROUND	1
1.3.1	Software Standards And Practices	1
1.3.2	Structured Analysis	2
1.3.3	Data Structure	2
1.3.4	Operator Interface	3
1.4	PROGRAM FUNCTIONS	3
1.4.1	Setup	4
1.4.2	Run	4
1.4.3	On-line Data Reduction	5
1.5	SCOPE OF THIS MANUAL	5
2.0	REFERENCED DOCUMENTS	6
2.1	SOFTWARE DEVELOPMENT TOOLS	6
2.2	SOFTWARE STANDARDS AND PRACTICES	6
2.3	INSTRUMENTATION MANUALS	6
2.4	VAX/VMS UTILITIES	6
3.0	GENERAL INFORMATION	7
3.1	EQUIPMENT CONFIGURATION	7
3.1.1	Computer	7
3.1.2	LV	7
3.1.3	Positioner	7
3.1.4	Operator's Console	7
3.2	COMPILATIONS AND ASSEMBLIES	7
4.0	SETUP	9
4.1	INITIAL SETTINGS	9
4.1.1	SUBROUTINE READ_TEST_SETUP (TEST)	9
4.1.2	SUBROUTINE READ_LV_SETUP (LV)	10
4.1.3	SUBROUTINE READ_TRAVERSE_SETUP (TRAVERSE)	11
4.2	SUBROUTINE DESCRIBE_TEST (TEST)	11
4.2.1	SUBROUTINE SET_TEST_NUMBER (KEYBOARD, ENTRY, NUMBER) ..	11
4.2.2	SUBROUTINE SET_TEST_FACILITY (KEYBOARD, ENTRY, FACILITY) ..	12
4.2.3	SUBROUTINE SET_RUN_NUMBER (KEYBOARD, ENTRY, NUMBER) ..	12
4.2.4	SUBROUTINE SET_CODE (KEYBOARD, ENTRY, NUMBER)	12
4.2.5	SUBROUTINE SET_RUN_TITLE (KEYBOARD, ENTRY, TITLE)	12
4.2.6	SUBROUTINE SET_RUN_ANGLE_OF_ATTACK (KEYBOARD, ENTRY, ALPHA) ..	12
4.2.7	SUBROUTINE SET_RUN_REFERENCE_VELOCITY (KEYBOARD, ENTRY, U0) ..	12
4.2.8	SUBROUTINE SET_RUN_REFERENCE_DISTANCE (KEYBOARD, ENTRY, D0) ..	12
4.2.9	SUBROUTINE UPDATE_TEST_SETUP (TEST)	13
4.3	SUBROUTINE SET_TRAVERSE (COMMAND_ENTRY, PASTEBOARD, POSITIONER, TRAVERSE) ..	13
4.3.1	SUBROUTINE SET_POSITIONING_MODE (PASTEBOARD, MODE)	13
4.3.2	SUBROUTINE UPDATE_TRAVERSE (TRAVERSE, POSITIONER)	14
4.4	SUBROUTINE SET_LV (INITIAL_COMMAND, PASTEBOARD, LV, LV_DATA) ..	14
4.4.1	SUBROUTINE DEFINE_LV_SYSTEM (PASTEBOARD, LV_SYSTEM, LV_-	

	DATA)14	
4.4.2	SUBROUTINE SET_LV_CONSTANTS (LV).....	18
4.4.3	SUBROUTINE SET_LV_STATS (PASTEBOARD, LV_STATS).....	19
4.4.4	SUBROUTINE UPDATE_LV_SETTINGS (LV).....	19
4.5	SUBROUTINE SET_POST_PROCESSING (POSTRUN).....	20
4.5.1	SUBROUTINE BASIC_STATS (LV_STATS)	20
5.0	RUN	21
5.1	CONDUCTING THE TEST.....	21
5.1.1	SUBROUTINE RUN_LOG (TEST_NUMBER).....	21
5.1.2	SUBROUTINE UPDATE_TEST (WHEN, TEST).....	21
5.1.3	SUBROUTINE GET_LV_DATA (LV, LV_TIME, LV_VELOCITY).....	21
6.0	ANALYZE	24
6.1	OFFLINE ANALYSIS.....	24
6.1.1	SUBROUTINE ANALYZE_TEST_DATA (TEST).....	24
7.0	UTILITIES.....	25
7.1	SCREEN CONTROL	25
7.1.1	SUBROUTINE SCREEN (OPERATION, PASTEBOARD_ID, MAIN, MAIN_- TITLE, MENU, MENU_TITLE, NUMBER, COMMANDS, COMMAND, KEYBOARD)25	
7.2	ERROR HANDLING	25
7.2.1	Subroutine ERROR_CHECK(ISTAT,IOSB,ERROR,DEVICE)	25
7.2.2	INTEGER*4 FUNCTION ERROR_HANDLER(SIGARGS,MECHARGS)	25
7.3	DEVICE CHANNEL ASSIGNMENT	26
7.3.1	SUBROUTINE ASSIGN_DEVICE (NAME, CHANNEL, ERROR).....	26

1.0 SCOPE

1.1 IDENTIFICATION

This software package is denoted as the Laser Velocimeter Data Analysis Package for the NASA Lewis Research Center and has the acronym LVDAP-LeRC. The current version is 1.0.

1.2 PURPOSE

This software package allows the operator to configure and acquire velocity data from a TSI LV system interfaced to a VAX through the Macrodyne LVABI interface to a DR11-W board or compatible. The TSI system consists of a 1990C Counter-type Signal Processor, an MI-990 multichannel interface, the LVABI interface from the MI-990 to the VAX, and the RMR-1989 rotating machinery resolver. Both the MI-990 and the RMR-1989 are configured by commands sent over an RS-232 serial port. This software supports up to three LV channels.

Designed to run on the Digital Equipment Corporation's family of VAX computers, the software package performs the following functions:

- o configures the LV system
- o reports and handles error conditions
- o acquires, reduces, stores and presents the LV velocities online

The remainder of this manual describes the components of the computer program that perform these functions. It also describes some of the features provided by the VAX/VMS operating system that assist in both the development and efficient operation of the software.

1.3 BACKGROUND

To provide insight into the rationale behind the LVDAP-LeRC computer program, this section presents the overall programming philosophy and development methods. Where practical the current LV software package incorporates modern software engineering practices. This section briefly describes the impact of these practices on the software development of the present computer program for the LV systems.

1.3.1 Software Standards And Practices

Recognizing the need for adequate software documentation, this software used a set of guidelines in documenting software requirements, design, code, testing, and operation. To avoid the voluminous reports often generated to satisfy documentation requirements, a structured, object-oriented approach was used in the development of the current LV software. With this approach the documentation can be broken down numerous small descriptions of the various modules that comprise the overall software package. This allows up-

dates or modifications to be restricted to small portions of the manual and prevents a major rewrite for each software revision.

The structured approach is possible only because of extensions to the FORTRAN-77 programming language supported by the Digital Equipment Corporation for the VAX series of computers. The two major features offered are the STRUCTURE statement for structuring and defining variables and the expansion in the size of subroutine and variable names from six to 32 characters. These features allow subroutine names to be more descriptive (CONDUCT_TEST rather than GDATA) and allows more lucid, structured variable naming conventions (LV.VELOCITY.SAMPLES rather than NUM).

1.3.2 Structured Analysis

For the current software, structured analysis consists of taking the three major functions - setup, data acquisition, and data analysis - and breaking them up into subfunctions. This process is repeated until the software tasks are partitioned into the smallest possible components. For instance, the setup function can be decomposed into defining the test, configuring the LV, and defining the traverse sequence. Configuring the LV consists of defining the conversion constants, setting up the LV electronics, and establishing the statistical environment, and so on. Data flow diagrams provide the means for keeping track of the decomposition and allow a means of defining subroutine names (for instance, SET_LV_INTERFACE for the subroutine that performs the function that defines the LV interface name and status).

1.3.3 Data Structure

At each stage of the functional decomposition the variables involved are defined in a data dictionary. Using the STRUCTURE statement in VAX FORTRAN (or the RECORD statement in Ada) is a convenient way of keeping track of these variables. It also leads to more recognizable variable names. Using the data flow diagrams along with the data dictionary allows the software developer to code the requirements in FORTRAN using the STRUCTURE statements for variable definition, the abbreviated subfunction definition for the subroutine name, and comments describing the function of the subroutine in more detail. This approach minimizes software development time, allows checking of the preliminary code from the requirements to the operational phases, and gives the programmer a way to code immediately in a structured manner, which is what many programmers do anyway, but without any overall plan.

In general, for most wind-tunnel applications, the data must be presented in three forms: instrument readings, physical quantities, and graphics. Initially, the data consists of instrument readings which may or may not be in the form of a physical quantity, such as position or velocity. This form is most convenient for a technician or operator to troubleshoot the instrument. Thus, the current LV software provides routines which acquire and present the raw instrument readings to facilitate instrument diagnostics during the setup procedure. For monitoring test conditions during the run, the software converts the instrument readings to physical quantities so that the operator can ensure proper test conditions during on-line acquisition. Finally, during off-line analysis, the data sets are combined to pro-

duce a composite graph of the data. This form allows conveyance of the maximum amount of information in the minimum amount of time.

1.3.4 Operator Interface

The operator interface relies heavily on the screen management facility introduced on the VAX. This facility supports multiple windows which allow command entries and menus describing the function of each command. Making use of the screen management facility allows the operator to concentrate on the test rather than memorize commands.

1.4 PROGRAM FUNCTIONS

The current LV program is titled LVDAP-LeRC. The function of this program is to set up, process and present fluid dynamics test data in a form suitable for the operator. The program supports the TSI laser velocimeter (LV) system, which furnishes velocity and time data. The software employs processing algorithms to derive the velocity statistics and to store and present these data for both on-line and off-line analysis.

The components of the TSI laser velocimeter system supported by the software are:

1. a model 1990C counter-type signal processor,
2. a model MI-990 multichannel interface, and
3. a model 1989 rotating machinery resolver.

The LV instrument readings enter the VAX through a LVABI/DR11-W Direct Memory Access (DMA) interface furnished by Macrodyne.

The LV system provides the following flow variables:

LV velocity data including

- number of samples,
- one, two, or three velocity components,
- means,
- standard deviations,
- histograms of each component;

time between samples;

angular position.

1.4.1 Setup

A functional description of the routines involved in setting up the LV and positioning instrumentation is given in Chapter 4. The following three sections give an overview of these functions.

1.4.1.1 Test Description

The first task of setting up for on-line data analysis is to identify the test. This consists of defining the test facility, assigning numbers for the test, run and code, and defining reference variables (length, velocity, angles).

1.4.1.2 Instrument Configuration

The next task during setup is to define and configure the instruments. These include the positioner and LV systems. The positioner consists of a Model 1989 Rotating Machinery Resolver (RMR) for detecting angular position. A sequence is then designated for data acquisition at selected angular points determined from the rotary encoder. The parameters necessary to convert the LV readings to velocity are also entered along with parameters relating to the velocity statistics.

1.4.1.3 Instrument Testing

To provide for troubleshooting and diagnostics, each instrument has a subroutine that tests the instrument. These routines allow the operator to send commands and receive data from the instrument to ensure proper operation prior to data acquisition. This capability will be included in future upgrades of the software in response to the requirements of the operating environment.

1.4.2 Run

Described in Chapter 5, the run function consists of synchronizing the data acquisition from the LV and angular positioning systems, performing on-line data analysis, storing the raw data, and presenting the results on-line on the operator's console. To perform these functions, the software relies heavily on the VAX/VMS operating system services. This function is currently supported by software developed at NASA for the LVABI. Some of the functions supported are described in the following sections.

1.4.2.1 Multiple Tasks

Since the LV system is connected to the VAX by a DR11-W interface, acquisition and processing of the data takes place simultaneously. This occurs because the DR11-W interface uses direct memory access (DMA), which transfers blocks of data via the hardware, not under software control. Not only are DMA transfers faster, but they free up the software for other tasks, such as reducing the data. In the present software, data is transferred into multiple buffers. As one buffer is being read into, another buffer is being processed.

1.4.2.2 Asynchronous Traps

Coordinating the transfers between multiple buffers requires asynchronous traps. Basically, these interrupt the data reduction process every time a buffer is full. The software can then choose to either stop or continue the data transfer depending on whether or not a sufficient number of samples have been acquired.

1.4.2.3 Operator Interrupt

Asynchronous traps from the operator's terminal allow interruption of the run sequence when the program is in the automatic positioning mode. This feature allows the operator to abort the run and return to the executive routine at his discretion.

1.4.2.4 Error Handling

Another important feature of the VAX/VMS operating system is the ability to detect and handle error conditions. In the present software, the source, description, and correction of the error are presented to the operator upon detection of an error.

1.4.3 On-line Data Reduction

The data reduction consists of converting the LV and RMR-1989 readings into velocity and position information. The velocity statistics provided are the number of samples, means, and standard deviations. This information is displayed on the operator's console. In addition to the statistics, the velocity histograms are also sent to a file for listing on a line printer.

Finally, the raw data along with test and instrument configurations are stored in a file for off-line data reduction.

1.5 SCOPE OF THIS MANUAL

The rest of this manual presents a description of the software and supporting information necessary to understand and modify the code. Chapter 2 gives documents which describe the software development techniques, VAX/VMS routines, and details of the instrumentation. Then Chapter 3 presents general information concerning the equipment configuration and operational structure of the software. Chapter 4 presents a functional breakdown of the setup function, which configures the LV prior to a test. Future data acquisition and reduction capabilities are presented in Chapter 5 based on anticipated requirements. Chapter 6 presents an overview of data analysis software, which is not currently supported but may be included in future updates as required. Finally, Chapter 7 presents some of the utility routines required for error handling and screen management.

2.0 REFERENCED DOCUMENTS

2.1 SOFTWARE DEVELOPMENT TOOLS

1. Booch, G., *Software Engineering with Ada*, 2nd Edition, Benjamin/Cummings Publishing Company, 1987
2. deMarco, T., *Structured Analysis and System Specification*, Yourdon, Inc, 1979
3. *Guide to Text Processing on VAX/VMS*, Digital Equipment Corporation, April 1986

2.2 SOFTWARE STANDARDS AND PRACTICES

1. *Guide to Creating Modular Procedures on VAX/VMS*, Digital Equipment Corporation, July 1985

2.3 INSTRUMENTATION MANUALS

1. *Model 1990C Counter-type Signal Processor, Instruction Manual*, TSI Inc., 500 Cardigan Road, P.O. Box 64394, St. Paul, MN 55164, (612)483-4711
2. *Model MI-990 Multichannel Interface, Instruction Manual*, TSI Inc., 500 Cardigan Road, P. O. Box 64394, St. Paul, MN 55164, (612)483-4711
3. *Model 1989 Rotating Machinery Resolver, Instruction Manual*, TSI Inc., 500 Cardigan Road, P. O. Box 64394, St. Paul, MN 55164, (612)483-4711
4. *Model 1999 Rotary Encoder, Instruction Manual*, TSI Inc., 500 Cardigan Road, P. O. Box 64394, St. Paul, MN 55164, (612)483-4711
5. *LVABI (Laser Velocimeter Autocovariance Buffer Interface) Instruction Manual*, Macrodyne, Inc., 4 Chelsea Place, P. O. Box 376, Clifton Park, NY 12065, (518)383-3800
6. *VAX/VMS I/O User's Reference Manual: Part I, Section 8, Terminal Driver*
7. *VAX/VMS I/O User's Reference Manual: Part II, Section 3, DR11-W and DRV11-WA Interface Driver*
8. *MLSI-DR11-W Plus Instruction Manual*, MDB Systems Inc., 1110 West Taft Ave., Box 5508, Orange CA 92613-5508, (714)998-6900

2.4 VAX/VMS UTILITIES

1. *Guide to Programming on VAX/VMS*, Digital Equipment Corporation, April 1986
2. *Programming in VAX FORTRAN*, AA-D034D-TE, Digital Equipment Corporation, September 1984

3.0 GENERAL INFORMATION

In the development of the software package LVDAP-LeRC, which operates a TSI LV system, specific hardware configurations are assumed. This chapter delineates these configurations.

3.1 EQUIPMENT CONFIGURATION

This section describes the equipment necessary for the software package LVDAP-LeRC to operate properly. It is broken down into the computer, LV and positioning subsystems, and the operator's console.

3.1.1 Computer

The software executes on one of the VAX series of computers manufactured by the Digital Equipment Corporation (DEC). The computers should use Version 4.0 or higher of the VAX/VMS operating system. For data storage, a disk system is required.

3.1.2 LV

This software supports the TSI LV systems. These systems consist of a 1990C counter-type signal processor, an MI-990 multichannel interface, an RMR-1989 rotating machinery resolver, and a DMA interface to the VAX by means of the Macrodyne LVABI interface.

3.1.3 Positioner

The positioning system supported consists of a Model 1999 rotary encoder controlled by a RMR-1989 rotating machinery resolver manufactured by TSI. The operating characteristics of these devices are given in References 3 and 4 of Section 2.3.

3.1.4 Operator's Console

The operator's console should be a VT100-type terminal or any other terminal compatible with the requirements of the VAX/VMS Screen Management Utility.

3.2 COMPILATIONS AND ASSEMBLIES

All software packages for LVDAP-LeRC are located in the subdirectory [.LERC-LV] under the object file library LEWIS-LV.OLB. The contents of the library can be viewed by invoking the VAX/VMS command `LIBRARY/LIST LEWIS-LV`. The main program resides in the source file `LERC-LV.FOR` and the object file is `LERC-LV.OBJ`. To assemble the program, first update the libraries. Then link using the VAX/VMS command

`$LINK LERC-LV, LEWIS-LV/L`

The main program, entitled `LERC_LV`, is described in Chapter 1 and calls the setup and run routines described in Chapters 4 and 5 using the utilities presented in Chapter 7.

4.0 SETUP

SUBROUTINE SETUP_TEST (INITIAL, LV_DATA) allows the operator to initialize the variables required to acquire and analyze information from the LV. The acquisition and analysis consist of acquiring the velocity data, presenting the velocity statistics online, and performing offline analysis of the results.

The setup consists of specifying the

- test identifiers (e.g. run, code),
- initial instrument settings, and
- parameter ranges and increments (e.g. position, velocity) varied during a run.

The offline postprocessing setup involves entering the

- type of postprocessing,
- identifying the data files, and
- processing parameters (samples, confidence limits, etc.).

The instruments supported are a

- positioner consisting of a TSI Model 1989 Rotating Machinery Resolver and a
- TSI laser velocimeter system by means of a multichannel interface, Model MI-990.

4.1 INITIAL SETTINGS

SUBROUTINE READ_SETUP_FILE - This routine reads the following setup files

1. TEST.SET - to obtain the default test description
2. LV.SET - for the LV settings
3. TRAVERSE.SET - to get traverse limits and increments

If the setup file exists, then the values read from that file form the initial settings; otherwise default values are assigned by the software.

4.1.1 SUBROUTINE READ_TEST_SETUP (TEST)

This routine reads the file TEST.SET, if available, to initialize the parameters identifying a test. If the file does not exist, then it is created, the initial values are set by the software, and are written to the file. The parameters set are

1. test facility,
2. test number,
3. run number,
4. run code number,

5. run title,
6. reference distances,
7. reference velocities, and
8. reference angles.

4.1.2 SUBROUTINE READ_LV_SETUP (LV)

This routine reads the file LV.SET to set the initial values for LV system operation and data reduction. If LV.SET does not already exist, then the LV parameters are set by this routine. There are three classes of LV settings - system, conversion constants, and statistics control. The parameters set are listed as follows.

System

1. Interface
2. Status
3. Clock frequency
4. Number of channels available
5. The status of each channel (on or off)
6. The number of fringes for each channel

Conversion Constants

1. Channel ID
2. Velocity units (English or Metric)
3. Fringe spacing
4. Velocity offsets introduced by the Bragg cell
5. Rotation angle for 3-D operation

Velocity Statistics Computation

1. Samples
2. On-line histograms
3. Histogram spacing
4. Confidence limits for bad point removal

4.1.3 SUBROUTINE READ_TRAVERSE_SETUP (TRAVERSE)

This operation reads the parameters that control the traverse sequence from the file TRAVERSE.SET. If the file does not exist or a read error occurs, the initial values are set by the program. The parameters initialized are

1. the traverse option (manual, operator controlled, or automatic);
2. the axis labels (x, y, z, u, v, w);
3. whether or not to increment the run after each axis change during automatic operation;
4. the rotation angle;
5. the starting point;
6. increment; and
7. number of increments.

4.2 SUBROUTINE DESCRIBE_TEST (TEST)

DESCRIBE_TEST allows the operator to enter the conditions that define a given test. The routine performs the following tasks:

1. initializes the test variables by reading the file TEST.SET and if the file does not exist, sets the test variables to default values;
2. allows the operator to enter parameters defining the test, which are
 - test number,
 - test facility,
 - run number,
 - run code,
 - run title,
 - reference distances,
 - reference velocities,
 - reference angles;
3. get the date and time from the VAX system clock.

4.2.1 SUBROUTINE SET_TEST_NUMBER (KEYBOARD, ENTRY, NUMBER)

This routine allows the operator to enter the test number. Valid entries are integers lying between 1 and 999, inclusive. The routine checks for a valid entry, and updates the test number value, if valid.

4.2.2 SUBROUTINE SET_TEST_FACILITY (KEYBOARD, ENTRY, FACILITY)

This routine allows the operator to enter the test facility. Valid facilities for LVDAP-LeRC are

- CE17,
- SE1,
- W7,
- UNK - UNKnown.

The entries are checked to make sure the facility name is valid before updating.

4.2.3 SUBROUTINE SET_RUN_NUMBER (KEYBOARD, ENTRY, NUMBER)

This routine allows the operator to enter the run number. Valid numbers are integers lying between 1 and 999, inclusive. The routine checks for a valid entry, and updates the run number value, if valid.

4.2.4 SUBROUTINE SET_CODE (KEYBOARD, ENTRY, NUMBER)

This routine allows the operator to enter the run code. Valid numbers are integers lying between 1 and 999, inclusive. The routine checks for a valid entry, and updates the run code value, if valid.

4.2.5 SUBROUTINE SET_RUN_TITLE (KEYBOARD, ENTRY, TITLE)

This routine allows the operator to enter the title of the run. The title can contain up to 60 characters.

4.2.6 SUBROUTINE SET_RUN_ANGLE_OF_ATTACK (KEYBOARD, ENTRY, ALPHA)

This routine allows the operator to enter the reference angles. Valid values of the 3 angles are real numbers lying between +180.0 and -180.0 degrees. The routine checks for a valid entry and updates the valid angles.

4.2.7 SUBROUTINE SET_RUN_REFERENCE_VELOCITY (KEYBOARD, ENTRY, U0)

This routine permits the operator to set U0, the reference velocity vector, for the test. Valid values of the 3 components of U0 are real numbers between 10,000 and -1000. The routine checks for a valid entry and updates U0 if valid.

4.2.8 SUBROUTINE SET_RUN_REFERENCE_DISTANCE (KEYBOARD,

ENTRY, D0)

This routine permits the operator to set D0, the reference distance, for the test. Valid values of D0 are real numbers between 10,000 and -1000. The routine checks for a valid entry and updates D0 if valid.

4.2.9 SUBROUTINE UPDATE_TEST_SETUP (TEST)

This routine updates the file TEST.SET upon completion of the setting of the test parameters.

4.3 SUBROUTINE SET_TRAVERSE (COMMAND_ENTRY, PASTEBOARD, POSITIONER, TRAVERSE)

This routine allows the operator to set up the initial conditions of a linear, three-axis positioner prior to running. The functions to be performed are:

1. noting whether the positioner is off-line, manually operated from the operator's console, or under automatic computer control;
2. designation of the number of steps per unit length for the stepper motors;
3. homing the y axis; and
4. establishing the traverse sequence.

The current version supports only the first function, with the positioner assumed set for manual operation only. Additional capability can be included as the need arises.

4.3.1 SUBROUTINE SET_POSITIONING_MODE (PASTEBOARD, MODE)

During a test sequence the operator needs to move the positioning system in one of three ways.

1. **Off-line operation through a joystick.** Here the computer does not control the operation of the positioning system. The operator should, however, be able to enter the position, if desired, to maintain a record of the position coordinates.
2. **Manually entered position.** In this mode the operator enters the desired position and the software sends the appropriate commands to move the positioner to that location. This mode moves the positioner one point at a time.
3. **Automatic traverse.** The automatic traverse mode notifies the system software that the positioning system is to step through a range of coordinates under computer control.

Currently, only the manual position is allowed and is set by the software.

4.3.2 SUBROUTINE UPDATE_TRAVERSE (TRAVERSE, POSITIONER)

This routine updates the file TRAVERSE.SET with the most recent changes in the traverse sequence.

4.4 SUBROUTINE SET_LV (INITIAL_COMMAND, PASTEBOARD, LV, LV_DATA)

This routine is responsible for defining and initializing the configuration of the LV system. This consists of three tasks:

1. define the LV electronics configuration through subroutine DEFINE_LV_SYSTEM,
2. specify the constants necessary to convert the LV readings to velocity (SET_LV_CONSTANTS), and
3. set the statistical constants for use in LV velocity processing (SET_LV_STATS)

The variables that define the LV system configuration are contained in the record LV defined in the data dictionary.

4.4.1 SUBROUTINE DEFINE_LV_SYSTEM (PASTEBOARD, LV_SYSTEM, LV_DATA)

This routine defines the following LV system parameters:

1. the LV interface,
2. the clock frequency,
3. the number of counter processors,
4. the counter processor mode of operation, and
5. the rotating machinery resolver (RMR) state.

4.4.1.1 SUBROUTINE SET_LV_INTERFACE (PASTEBOARD, INTERFACE)

This software module allows the operator to configure the TSI MI-990 multichannel LV interface. The routine then assigns and allocates the interface to ensure proper operation.

4.4.1.2 SUBROUTINE SET_LV_CLOCK (PASTEBOARD, CLOCK_FREQUENCY)

SET_LV_CLOCK allows the operator to enter the clock frequency of the LV counter-processor system. The input is in megaHertz and the operator entry is validated.

4.4.1.3 SUBROUTINE SET_LV_COUNTERS (PASTEBOARD, LV)

This module specifies the number of TSI 1990C counter processors connected to the MI-990 multichannel interface. There can be up to 3 processors. The routine sets the parameter LV.SYSTEM.CHANNEL.AVAILABLE to the number of processors and then issues the command to the MI-990 that indicates the number of processors available.

4.4.1.4 SUBROUTINE SET_LV_MODE (PASTEBOARD, LV)

SET_LV_MODE allows the LV operator to set the TSI Model MI-990 multichannel interface module. The operator can control the MI-990 through the following commands:

- CO - allow only coincident data taken within a specified time interval;
- CW - set the number of C words;
- ET - set for even time sampling only;
- IT - inhibit transfer of the time between samples;
- RS - allow random sampling, i.e., when any counter acquires a validated data point, take the point and reset;
- TB - send the time between samples along with the fringe crossing times.

4.4.1.4.1 SUBROUTINE SET_LV_COINCIDENCE (PASTEBOARD, LV)

This routine sets the TSI MI-990 LV interface so that data is taken only when all counters have valid data within a specified time interval. This routine allows the operator to set this time interval. The following parameters are set.

LV.MI.CONCIDENCE.STATUS - 'ON' or 'OFF'

LV.MI.CONCIDENCE.MANTISSA - a valid number within the interval $(0,9) = m$

LV.MI.CONCIDENCE.EXPONENT - a valid number within the interval $(0,6) = e$

where the coincidence time delay t is $t = m^{e-6}$ seconds.

4.4.1.4.2 SUBROUTINE SET_LV_CWORD (PASTEBOARD, LV)

This module specifies the number of C-words that the MI-990 transfers after acquiring LV readings from the counter processors. The C-word status is set to either on or off and the number of C-words can be set to between 0 and 15.

4.4.1.4.3 SUBROUTINE SET_LV_ETS (PASTEBOARD, LV)

This module sets the TSI MI-990 LV interface to sample the data at equal time intervals. The interval should be greater than the computer's sampling time. The Equal Time status (ETS) can be set either on or off and the time interval can be set by entering a mantissa m

and exponent e . Then the time interval t is m^{e-6} seconds, where e lies in the interval (0,6) and m falls on or between (0,9).

4.4.1.4.4 SUBROUTINE SET_LV_INHIBIT (PASTEBOARD, LV)

This module prevents the TSI MI-990 LV interface from sending the time between samples. SET_LV_INHIBIT allows the operator to turn the inhibit on or off, issues the appropriate command to the LV interface, and sets the equal time status and time between data status to off if the inhibit status is on.

4.4.1.4.5 SUBROUTINE SET_LV_RANDOM (PASTEBOARD, LV)

This routine allows the TSI MI-990 LV interface to acquire the fringe crossing times whenever any LV channel has valid data. This mode is in contrast to the coincidence mode, where all channels must have valid data before transfers can occur.

4.4.1.4.6 SUBROUTINE SET_LV_TBD (PASTEBOARD, LV)

SET_LV_TBD allows the TSI MI-990 LV interface to send the time between samples along with the fringe crossing times. This feature can be inhibited using the IT mode command, which calls the subroutine SET_LV_INHIBIT.

4.4.1.4.7 SUBROUTINE SET_MI_READ (LV)

SET_MI_READ reads the TSI MI-990 multichannel interface after a command is sent. If the command is accepted, it is echoed back. If an error occurs, the interface returns the string ERR<CR><LF> and the routine issues an error flag.

4.4.1.5 SUBROUTINE SET_LV_RMR (PASTEBOARD, LV)

This routine allows the operator to set up the TSI Model 1989 RMR Rotating Machinery Resolver. The operator can enter the following commands.

AD - Set the addresses if the RMR memory locations;

IF - Define the RMR interface;

LD - Lock the detector sensitivity to one of the following - 96, 192, 384, 768 arc minutes

MC - Set the machine cycle to 360 or 720 degrees per revolution

RM - Set the RMR mode to one of the following:

PLL - phase-locked loop,

Shaft encoder x2 mode,

Shaft encoder x4 mode, or

Micro mode;

TC - Toggle machine cycle.

The program issues the command descriptions to the operator, validates the command from the operator, and performs the indicated operation.

4.4.1.5.1 SUBROUTINE SET_LV_RMR_IF (PASTEBOARD, INTERFACE)

This module allows the operator to specify the TSI RMR-1989 interface. The routine then assigns and allocates the interface to ensure proper operation or notifies the operator of an error condition.

4.4.1.5.2 SUBROUTINE SET_LV_RMR_CYCLE (PASTEBOARD, RMR)

SET_LV_RMR_CYCLE sets the machine cycle parameter RMR.CYCLE of the RMR 1989 to 0 for 720 degrees per cycle and 1 for 360.

4.4.1.5.3 SUBROUTINE SET_LV_RMR_LOCK (PASTEBOARD, RMR)

This module sets the sensitivity of the TSI RMR 1989 to either 96, 192, 384, or 768 arc minutes and stores the setting in the parameter RMR.LOCK.

4.4.1.5.4 SUBROUTINE SET_LV_RMR_MEMORY (PASTEBOARD, RMR)

This module controls the memory settings of the RMR-1989. The memory addresses are configured so that each corresponds to 0.1 degree of the 360 degree cycle in the phased-locked loop (PLL) mode. By setting memory locations to zero, regions occupied by obstructions such as rotating blades can be blanked out during the cycle. Commands are entered in the following order:

1. Set/clear the memory segment
2. Starting address
3. Ending address

Memory addresses fall between 0 and 4095 for the first 360 degrees and from 4096 to 8191 for 360 to 720 degrees.

4.4.1.5.5 SUBROUTINE SET_RMR_MODE (PASTEBOARD, RMR)

This module sets the mode of operation for the RMR by setting the control word to one of the following values:

- 0 for phase-locked loop (PLL),
- 1 for shaft-encode x2,
- 2 for shaft-encode x4, and
- 3 to set micro mode.

4.4.1.5.6 SUBROUTINE SET_LV_RMR_TOGGLE (PASTEBOARD, RMR)

This routine applies when the machine cycle is in the 720 degree setting. The resolver cannot know which part of the cycle is the first half and which is the second. This routine allows the operator to set the correct half-cycle.

4.4.2 SUBROUTINE SET_LV_CONSTANTS (LV)

This routine sets the constants necessary to convert the LV fringe crossing time word to velocity and to identify the velocity component. To do so requires the following tasks:

1. assign the velocity components to the LV channels,
2. input the fringe spacing and sign,
3. compute conversion factors, and
4. set velocity offsets.

4.4.2.1 SUBROUTINE ASSIGN_COMPONENTS (PASTEBOARD, LV)

This routine allows the operator to identify the velocity components measured by each of the LV channels. The identifier consists of two characters. The first identifies the velocity component (U, V, or W) and the second can be 1 or 2 for spacial correlations or a space when the LV fringe pattern aligns with the velocity component directly.

For 3-D measurements the second character is the letter U, V, or W, which, together with the first character, denotes the velocity components measured by a single set of fringes. When more than one component per fringe is measured, the program requests the angle between the optical axis and the component denoted by the first character. This information then allows computation of the component denoted by the second character.

4.4.2.2 SUBROUTINE SET_FRINGE_SPACING (PASTEBOARD, LV)

Set_Fringe_Spacing allows the operator to enter the fringe spacing for up to eight channels of LV data. If desired, the operator may enter the beam spacing, focal length, and wavelength and let the software compute the spacing.

4.4.2.3 SUBROUTINE LV_CONVERSION_FACTOR (PASTEBOARD, LV)

This routine allows the operator to select the units of velocity in either meters or feet per second. When the operator enters an 'M', the units are in metric and when he enters an 'E', English units are indicated.

4.4.2.4 SUBROUTINE LV_VELOCITY_OFFSETS (PASTEBOARD, LV)

The use of a Bragg cell introduces an offset in the mean velocity values. LV_VELOCITY_OFFSETS allows the operator to enter the offset velocities.

4.4.3 SUBROUTINE SET_LV_STATS (PASTEBOARD, LV_STATS)

This routine provides control of the LV statistics. It allows the operator to set the following:

1. the number of data samples,
2. the histogram spacing, and
3. the confidence limit for histogram clipping.

These variables have a direct impact on the computation of the velocity statistics.

4.4.3.1 SUBROUTINE SET_SAMPLES (ENTRY, KEYBOARD, SAMPLES)

Set samples allows the operator to specify the number of samples to be acquired for an instrument during a run.

4.4.3.2 SUBROUTINE SET_HISTOGRAM_LIMITS (ENTRY, KEYBOARD, CONFIDENCE_LIMIT)

For histogram clipping, this routine sets the confidence limits in per cent. Data falling outside these limits are ignored, i. e. clipped.

4.4.3.3 SUBROUTINE SET_HISTOGRAM_SPACING (ENTRY, KEYBOARD, SPACING)

SET_HISTOGRAM_SPACING allows the operator to enter the desired histogram spacing.

4.4.3.4 SUBROUTINE SET_HISTOGRAM_OPTION (ENTRY, KEYBOARD, OPTION)

This routine allows the operator to specify whether or not he wants to see plots of the histogram on-line. If so, he can manually clip the histograms.

4.4.4 SUBROUTINE UPDATE_LV_SETTINGS (LV)

This routine reads the file LV.SET to set the initial values for LV system operation and data reduction. If LV.SET does not already exist, then the LV parameters are set by this routine. There are three classes of LV settings - system, conversion constants, and statistics control. The parameters set are given in the following list.

System

1. Interface
2. Status
3. Clock frequency

4. Number of channels available
5. The status of each channel (on or off)
6. The number of fringes for each channel
7. The velocity conversion factor

Conversion Constants

1. Channel ID
2. Velocity units (English or Metric)
3. Fringe spacing
4. Velocity offsets introduced by the Bragg cell
5. Rotation angle for 3-D operation

Velocity Statistics Computation

1. Samples
2. On-line histograms
3. Histogram spacing
4. Confidence limits for bad point removal

4.5 SUBROUTINE SET_POST_PROCESSING (POSTRUN)

This routine allows the operator to specify the type of post processing desired for the LV data desired. This is a future option.

4.5.1 SUBROUTINE BASIC_STATS (LV_STATS)

This routine will be included as a future upgrade and will allow the operator to enter the parameters required to compute the basic velocity statistics from LV data. These will include:

1. the test number,
2. initial and final runs and codes,
3. reference distances and velocities,
4. histogram spacing and confidence limits for clipping, and
5. bias correction.

Each entry will be validated to ensure proper operation.

5.0 RUN

This section gives an overview the functions that must be performed by the software during a run. These functions are planned for future upgrades pending final requirements definition.

5.1 CONDUCTING THE TEST

SUBROUTINE CONDUCT_TEST (CURRENT, ACQUIRED) is responsible for acquiring and reducing data from the TSI LV system over a range of positions. The tasks perform the following functions:

1. acquire and reduce the LV data and
2. present the results on-line.

5.1.1 SUBROUTINE RUN_LOG (TEST_NUMBER)

This routine opens the run log to record the run, code, facility, title, date, and start and stop times. The title of the run log is XXX.LOG where xxx is the TEST_NUMBER.

5.1.2 SUBROUTINE UPDATE_TEST (WHEN, TEST)

This routine updates the test variables prior to acquisition of LV or positioner data. This update consists of

1. updating the current date and start time;
2. incrementing the code, or if this is a new run, setting the code to 1; and
3. updating the run log file.

5.1.3 SUBROUTINE GET_LV_DATA (LV, LV_TIME, LV_VELOCITY)

This routine is currently not used. If required, its functions are to

1. transfer data from the TSI LV system via the LVABI;
2. coordinate transfer/processing tasks by means of asynchronous traps;
3. call CONVERT_LV_DATA to obtain velocity statistics; and
4. compute the histogram limits.

These functions are currently assumed to be supported by software developed within NASA for controlling the LVABI interface between the TSI LV system and the DR11 interface to the VAX.

5.1.3.1 SUBROUTINE OPEN_LV_FILE (FILE_TYPE, FORM, RUN_NUMBER, RUN_CODE)

This routine opens the data files for the LV. The name of the files have the form

xxittccc.dat

where

xx = blank for the raw data file,
= UC for the file containing the listing of the printout of the unclipped LV velocity data,
itt = run number, and
ccc = code.

5.1.3.2 SUBROUTINE HEADER_FILE (FILE_TYPE, OPERATION, LV, LV_VELOCITY, POSITION, TEST, TRAVERSE)

HEADER_FILE performs read/write operations to update the header file. The parameter OPERATION can assume the following settings:

‘READ’ to read the header file,
‘WRITE’ to write to the header file, or
‘ERROR’ signals an error condition.

5.1.3.3 SUBROUTINE CONVERT_LV_DATA (BUFFER, OPERATION, LV, LV_TIME, LV_VELOCITY)

This routine converts the data words from the LV system into physical quantities. The word contains

1. time between samples and
2. fringe crossing time.

The data are unpacked, the velocity computed, and the velocity statistics determined. The unpacked data are compressed and stored on disk for later data reduction. This routine will be tailored to the needs of the test operator as these become more clearly defined.

5.1.3.4 SUBROUTINE AUTO_CLIP (SPACING, CONFIDENCE_LIMIT, UPPER_LIMIT, LOWER_LIMIT, HISTOGRAM)

AUTO_CLIP eliminates points from a histogram that lie outside the specified confidence limits. It then determines the range of parameter values based on the velocity data that lie between these limits.

5.1.3.5 SUBROUTINE PRESENT_ONLINE_RESULTS (PASTEBOARD, TEST, POSITIONER, TRAVERSE, LV_VELOCITY)

PRESENT_ONLINE_RESULTS writes the run and code, position, and velocity samples, means, and standard deviations to the operator's console.

5.1.3.6 SUBROUTINE LIST_ONLINE_RESULTS (TEST, POSITIONER, SPACING, LV_VELOCITY)

LIST_ONLINE_RESULTS presents the means, standard deviations, and histograms of velocity data to an ASCII file suitable for presentation on a line printer. The data are unclipped.

6.0 ANALYZE

A primary function of the LV software is data analysis. This software module allows the operator to perform more extensive processing of the LV velocity data offline in addition to the online capability. These routines will be added as required to support the TSI LV system.

6.1 OFFLINE ANALYSIS

After the LV data is acquired online, it is previewed and stored for more detailed data analysis. This capability is planned as a future enhancement and supports the functions given in the next section.

6.1.1 SUBROUTINE ANALYZE_TEST_DATA (TEST)

This package of routines will analyze the test data acquired with the LV system. The tasks to be supported are

1. basic velocity statistics including bias removal and higher order moments through kurtosis,
2. time series analysis,
3. conditional sampling,
4. profile drag from wake surveys, and
5. vortex drag from the transverse components of velocity in the wake.

The software modules for each of these functions currently exist and need to be integrated into the overall LVDAP-LeRC data analysis package.

7.0 UTILITIES

Two classes of utilities are discussed in this section. The first is the screen display and the second is error handling capabilities. Both of these functions use VAX/VMS library routines to carry out the necessary operations.

7.1 SCREEN CONTROL

This function employs the VAX/VMS Screen Management (SMG) utility to present the operator with interactive menus and online data displays.

7.1.1 SUBROUTINE SCREEN (OPERATION, PASTEBOARD_ID, MAIN, MAIN_TITLE, MENU, MENU_TITLE, NUMBER, COMMANDS, COMMAND, KEYBOARD)

This routine controls the display on the operator's terminal during setup. It creates the following displays:

1. the main display containing the title of the screen and the main border;
2. the menu, which displays the allowable commands and their functions; and
3. the command entry overlay, which defines the borders for the screen for command entry.

7.2 ERROR HANDLING

During a run it is important for the operator to be aware of any error condition and its source so that corrective action can be initiated. The VAX/VMS operating system provides a wide range of error checking and handling utilities, which are applied here.

7.2.1 Subroutine ERROR_CHECK(ISTAT,IOSB,ERROR,DEVICE)

This is a custom-written error handling routine that provides the following information:

- source of the error,
- description of the error condition, and
- action to be taken for error recovery.

All error messages are written to file FOR013.DAT, which is read and then deleted after print out of the appropriate error messages resulting from the indicated error condition.

7.2.2 INTEGER*4 FUNCTION ERROR_HANDLER(SIGARGS,MECHARGS)

This routine intercepts errors which occur during I/O from the various devices involved. For recognized errors, a message is printed denoting the source, error condition, and possible corrective action.

7.3 DEVICE CHANNEL ASSIGNMENT

Prior to any data transfers to or from an instrument, the channel must be assigned and allocated to the process running the LVDAP-LeRC software. This assignment prevents possible conflicts when more than one task compete for the instrument.

7.3.1 SUBROUTINE ASSIGN_DEVICE (NAME, CHANNEL, ERROR)

This routine assigns and allocates a device during runtime using the VAX/VMS system services `SY$ASSIGN` and `SY$ALLOC`. Any errors are handled through the routine `ERROR_CHECK`.

6.2 User's Manual

The user's manual presents the instructions for properly operating the LV software developed during 1991 under NASA Grant NASG3-1215.

1.0	SCOPE	1
1.1	IDENTIFICATION	1
1.2	PURPOSE	1
2.0	REFERENCED DOCUMENTS	2
2.1	INSTRUMENTATION MANUALS	2
2.2	VAX/VMS UTILITIES	2
2.3	INSTRUMENTATION MANUALS	2
3.0	INSTRUCTIONS FOR USE	3
3.1	STARTUP	3
3.2	COMMAND STRUCTURE	3
3.3	SETUP	5
3.3.1	LV	6
3.3.2	Test	34
3.3.3	Traverse	36
3.4	RUN	37
3.5	SIMULATE A TEST	39
4.0	ERRORS	40

1.0 SCOPE

1.1 IDENTIFICATION

The identification nomenclature of this software package is LVDAP-LeRC, Version 1.0. The LVDAP stands for the Laser Velocimeter Data Acquisition Program, which is the function of this software. The LeRC is NASA's Lewis Research Center, which performs research and development into aerospace propulsion systems. Consequently, a goal of LVDAP-LeRC is to support NASA-Lewis in its efforts. This first version is concerned mainly with the proper configuration of the LV system prior to a test. Subsequent versions will enhance the setup function and add data analysis and presentation capabilities as well.

1.2 PURPOSE

This software program allows the operator to acquire, reduce, and preview velocity data from the LV systems designed and built by Thermal Systems Incorporated (TSI). These systems consist of the optical and electronic LV components for velocity measurement and a rotating machinery resolver for measuring angular position. This software supports up to eight LV channels and a Rotating Machinery Resolver (RMR) model RMR-1989.

Designed to run on the Digital Equipment Corporation's family of VAX computers, the software package performs the following functions:

- configuration of the positioning and LV systems
- testing of individual system components
- error reporting and handling
- data acquisition, online reduction, storage, and presentation

The remainder of this manual describes the procedure for executing the computer program that performs these functions. It also presents the commands necessary to conduct a wind tunnel test using the LV and positioning systems along with a description of the input and output generated. Finally, it presents some of the error and diagnostic tools for monitoring system performance.

2.0 REFERENCED DOCUMENTS

2.1 INSTRUMENTATION MANUALS

1. VAX/VMS I/O User's Reference Manual: Part I, Section 8, Terminal Driver
2. VAX/VMS I/O User's Reference Manual: Part II, Section 3, DR11-W and DRV11-WA Interface Driver

2.2 VAX/VMS UTILITIES

1. Guide to Programming on VAX/VMS, Digital Equipment Corporation, April 1986
2. Programming in VAX FORTRAN, Digital Equipment Corporation, September 1984

3.0 INSTRUCTIONS FOR USE

The program operates on the VAX family of computers. It has been developed and tested on the VAX 11/7xx and VAXStation II/GPX series. The hardware supported consists of the TSI laser velocimeter (LV) systems, which includes an MI-990 multichannel interface, a RMR-1989 rotating machinery encoder, and a TSI- 1990 counter processor. The peripherals required are a hard disk and an operator's terminal compatible with the VAX VMS screen management system. A tape drive or optical disk, while not mandatory, allows backup or storage the data.

3.1 STARTUP

To begin execution, type \$RUN LERC-LV. Upon initiation, the program reads the setup files, which contain the configuration information stored from previous executions. If these files do not exist, they are created during the setup procedure and are of type SET. These files are updated, as required, during subsequent setup procedures.

After program execution, two files are generated for each run and code. Their names will have the form rrrccc.DAT and UCrrrccc.DAT where rrr = run number and ccc = code. The first file contains the raw LV data suitable for reduction using offline data analysis software. The second contains the unclipped velocity statistics suitable for listing on a line printer. In addition, the run, code, facility, run title, and start and stop times for each data set are stored in an ASCII file with the name ttt.LOG, where ttt is the test number. The final file generated is an ASCII file entitled POSITION.TABLE. This file is only generated during automatic positioning and is read sequentially to determine the positioning sequence. This file can be generated online during program execution or offline.

3.2 COMMAND STRUCTURE

Upon program execution the menu shown in Figure 3-1 appears on the screen. The commands currently supported are SE (SEtup), RU (RUn), ST (Simulate a Test) and EN (ENd). AN (ANalyze) and RR (Run and Reduce) are included for future upgrades. SE allows modification of the test and instrument parameters; whereas RU sets the position and acquires, analyzes, and displays the LV velocity data. ST simulates a test using the TSI LV system and allows analysis of the instrument resolution and software checkout.

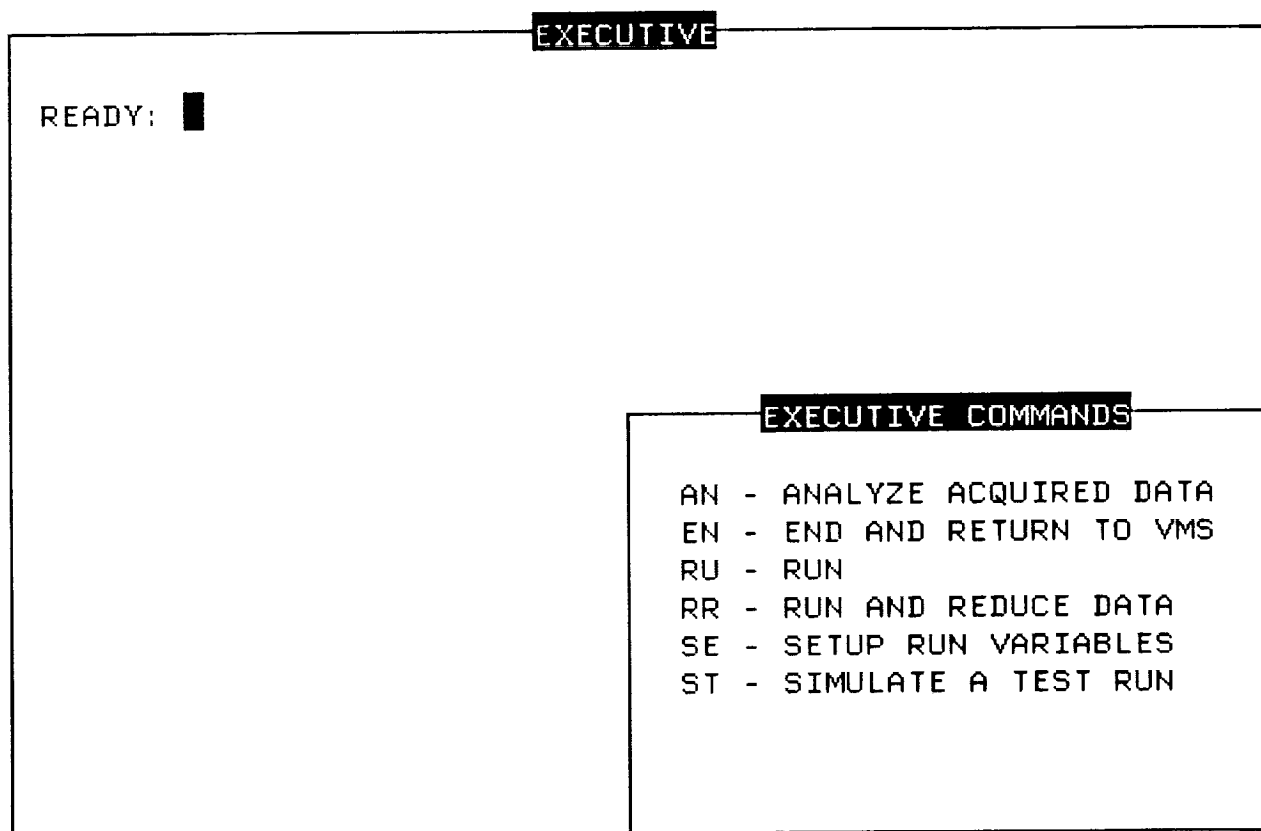


Figure 3-1. Executive (or Main) Menu

3.3 SETUP

SE. Entering SE after the READY: prompt causes the menu shown in Figure 3-2 to appear on the screen. The commands located in the submenu in the lower right-hand corner of the screen provides for configuration of the LV, test description, and traverse sequence. The traverse option is included for future expansion incorporating a positioning sequence. Throughout the setup procedure, entering a carriage return or blanks retains the displayed values. Invalid entries result in again displaying the current values for reentry. This traverse option allows future expansion with a linear positioning system.

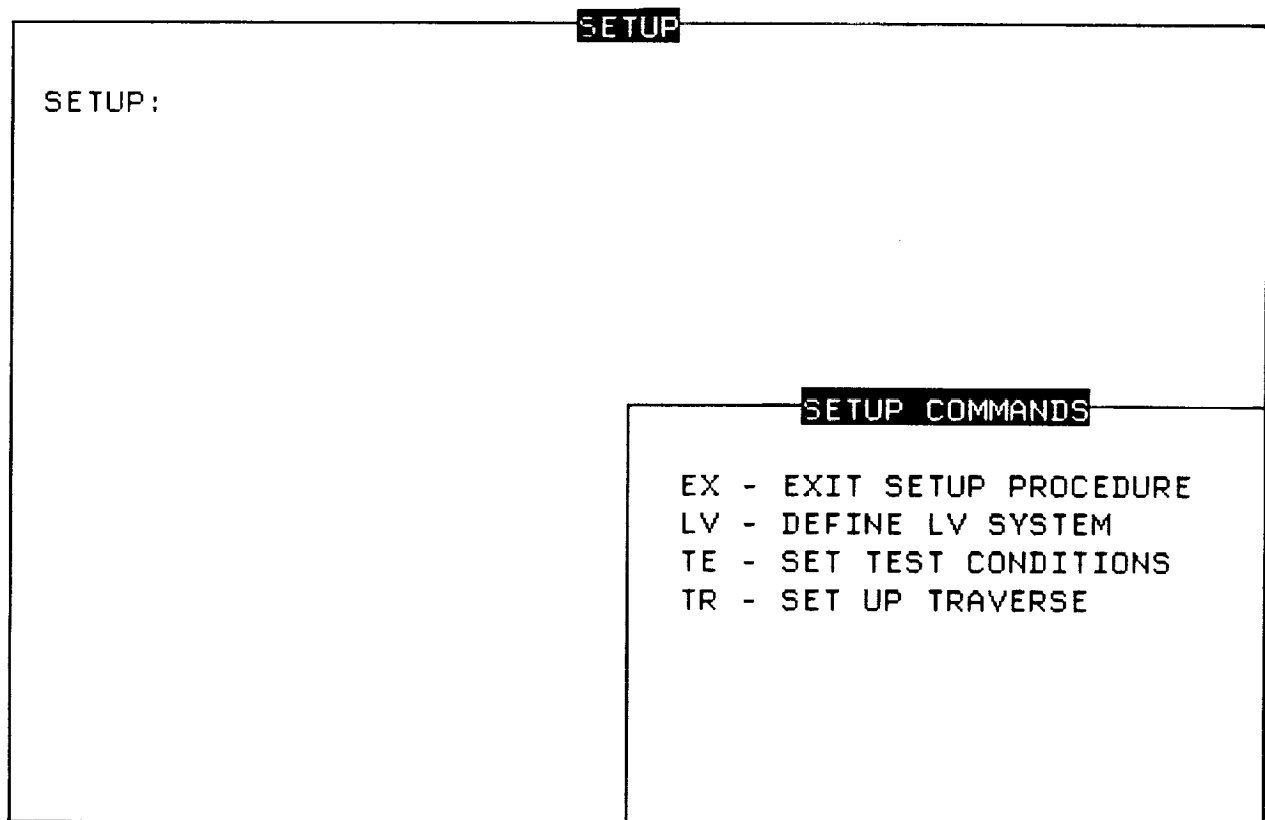


Figure 3-2. Setup Menu

3.3.1 LV

LV. Entering the command LV after the SETUP: prompt on the setup menu (or SE/LV after the READY: prompt on the executive menu) produces the menu shown in Figure 3-3. Here, the commands necessary to convert the raw LV data into appropriate velocity units (CC), to define the LV interface configuration (DL), and to control the velocity statistics (SS) can be entered.

LV SETUP						
LV: EX						
LV: <input type="text"/>						
	<table border="1"><thead><tr><th>LV COMMANDS</th></tr></thead><tbody><tr><td>CC - SET CONVERSION CONSTANTS</td></tr><tr><td>DL - DEFINE LV SYSTEM</td></tr><tr><td>SS - SET STATISTICAL VARIABLES</td></tr><tr><td>EX - EXIT LV SETUP</td></tr></tbody></table>	LV COMMANDS	CC - SET CONVERSION CONSTANTS	DL - DEFINE LV SYSTEM	SS - SET STATISTICAL VARIABLES	EX - EXIT LV SETUP
LV COMMANDS						
CC - SET CONVERSION CONSTANTS						
DL - DEFINE LV SYSTEM						
SS - SET STATISTICAL VARIABLES						
EX - EXIT LV SETUP						

Figure 3-3. Menu of Commands for Configuring the LV.

3.3.1.1 Setting The Conversion Constants

CC. Upon entering the command CC after the prompt LV: on the LV setup menu (or the command string SE/LV/CC from the executive menu), the screen displays the data shown in Figure 3-4. To enter a channel ID, position the cursor under the desired channel or channels and enter a valid identification for the velocity component. For the 3-D LV configuration, enter the angle between the green and blue beams for channel 3 as well as a channel ID of UW for channel 3. A nonzero angle flags the data reduction routine that 3-D data is available. Invalid entries result in a redisplay of the current settings for reentry.

LV CHANNEL IDENTIFICATION		
CHANNEL	1	2
ID	U	V
CHANNEL	1	2
ANGLE	0.0	0.0

VALID ENTRIES
A VALID CHANNEL ID IS
U, V, W, UW, VW, U2, V2, W2
VALID ANGLES LIE BETWEEN
-180.0 AND +180.0 DEGREES

Figure 3-4. Assigning Velocity Components and Angles to the LV Channels

After channel identification, the display shown in Figure 3-5 appears. Here, you have an option of either entering the beam spacing directly (D), or computing the fringe spacing from entered values of beam spacing, focal length, and wavelength, as in Figure 3-5b.

LV FRINGE SPACING			
OPTION	D		
FRINGE SPACE	15.00	14.00	
	16.	15.	
FRINGE SPACE	16.00	15.00	
	15.	14.	
FRINGE SPACE	15.00	14.00	

DESCRIPTION OF ENTRIES
OPTIONS - ENTER : D - TO ENTER FRINGE SPACING DIRECTLY C - TO COMPUTE SPACING FROM THE BEAM SPACING, FOCAL LENGTH, AND WAVELENGTH, IN NANOMETERS. THE BEAM SPACING AND FOCAL LENGTH UNITS MUST AGREE

Figure 3-5a. Direct Entry of Fringe Spacing

LV FRINGE SPACING

OPTION C

BEAM SPACING 1.00 1.00

LV FRINGE SPACING

FOCAL LENGTH 30.00 30.00

WAVELENGTH 514.50 514.50
 488.

WAVELENGTH 514.50 488.00

FRINGE SPACE 15.44 14.64

ENTER <CR> TO EXIT

DESCRIPTION OF ENTRIES

OPTIONS - ENTER :

D - TO ENTER FRINGE SPACING
DIRECTLY

C - TO COMPUTE SPACING FROM
THE BEAM SPACING, FOCAL
LENGTH, AND WAVELENGTH,
IN NANOMETERS. THE BEAM
SPACING AND FOCAL
LENGTH UNITS MUST AGREE

Figure 3-5b. Computed Fringe Spacing

Next, the velocity units are requested as shown in Figure 3-6. The velocity can be expressed in either feet (EN) or meters (ME) per second.

LV VELOCITY UNITS	
UNITS: ME	
UNITS: ME	
EN	
UNITS: EN	
ME	
UNITS: ME	

VALID UNITS
VALID ENTRIES FOR VELOCITY UNITS ARE
EN - FOR ENGLISH UNITS (FEET PER SECOND)
ME - FOR METRIC UNITS (METERS PER SECOND)

Figure 3-6. Velocity Units Entry

The final entries required for converting raw LV time words into velocity data are the velocity offsets for each channel. As shown in Figure 3-7, these offsets are entered by positioning the cursor to the appropriate channel or channels and entering the offsets. Illegal entries result in the current value being maintained and redisplayed for reentry. The entries are in either meters (MPS) or feet (FPS) per second, depending on whether metric or English units have been selected in Figure 3-6. After entering the velocity offsets the program returns to the LV setup menu (or the executive menu if the command SE/LV/CC is used.)

LV VELOCITY OFFSETS		
OFFSETS-MPS	50.00	55.00
OFFSETS-MPS	50.00	55.00
OFFSETS-MPS	50.00	55.00
	100.	110.
OFFSETS-MPS	100.00	110.00

DESCRIPTION OF ENTRIES

ENTER THE VELOCITY OFFSETS
CAUSED BY THE BRAGG CELL.
THE VELOCITY UNITS SHOULD
CORRESPOND TO THE ENGLISH
OR METRIC UNITS SELECTED.

Figure 3-7. Velocity Offset Entries

3.3.1.2 Defining The LV System Configuration

DL. To define the LV system configuration, the DL (Define LV system) command is entered in response to the LV: prompt on the LV SETUP menu shown in Figure 3-8. Configuring the LV can also be initiated from the executive menu using the command string SE/LV/DL after the READY: prompt.

EXECUTIVE	
READY: SE	
READY: SE/LV	
READY: SE/LV/CC	
READY: SE/LV/CC	
READY: SE/LV/CC	
READY: SE/LV/CC	
READY: SE/LV/DL	
READY: SE/LV/DL	

EXECUTIVE COMMANDS
AN - ANALYZE ACQUIRED DATA
EN - END AND RETURN TO VMS
RU - RUN
RR - RUN AND REDUCE DATA
SE - SETUP RUN VARIABLES
ST - SIMULATE A TEST RUN

Figure 3-8. LV Setup Command for Defining the LV System

After issuing the DL (Define LV) command, the menu entitled DEFINE LV SYSTEM VARIABLES appears on the screen as shown in Figure 3-9. As shown in the lower right-hand corner of the menu, valid commands consist of CF, EX, IF, SC, SM and SR. These commands serve to define the LV system configuration.

DEFINE LV SYSTEM VARIABLES								
LV SYSTEM: █								
	<table border="1"><thead><tr><th>LV SYSTEM COMMANDS</th></tr></thead><tbody><tr><td>CF - SET CLOCK FREQUENCY, IN MHZ</td></tr><tr><td>EX - EXIT LV SYSTEM SETUP</td></tr><tr><td>IF - DEFINE LV INTERFACE</td></tr><tr><td>SC - SET NUMBER OF LV COUNTERS</td></tr><tr><td>SM - SET MSI-990 MODE</td></tr><tr><td>SR - SETUP RMR-1989</td></tr></tbody></table>	LV SYSTEM COMMANDS	CF - SET CLOCK FREQUENCY, IN MHZ	EX - EXIT LV SYSTEM SETUP	IF - DEFINE LV INTERFACE	SC - SET NUMBER OF LV COUNTERS	SM - SET MSI-990 MODE	SR - SETUP RMR-1989
LV SYSTEM COMMANDS								
CF - SET CLOCK FREQUENCY, IN MHZ								
EX - EXIT LV SYSTEM SETUP								
IF - DEFINE LV INTERFACE								
SC - SET NUMBER OF LV COUNTERS								
SM - SET MSI-990 MODE								
SR - SETUP RMR-1989								

Figure 3-9. The Menu for Defining the LV System Variables

CF. Entering the CF (Clock Frequency) command after the LV SYSTEM: prompt allows the operator to enter the value of the frequency of the LV system clock, in megaHertz (MHz), as shown in Figure 3-10. With the current systems, this value is 1000.

LV CLOCK FREQUENCY	
CLOCK FREQUENCY	1000
	500
CLOCK FREQUENCY	500
	1000
CLOCK FREQUENCY	1000
	█

VALID ENTRY
THE CLOCK FREQUENCY IS AN INTEGER BETWEEN 1 AND 10,000 MHZ

Figure 3-10. Entering the LV System Clock Frequency

IF. Entering the IF (InterFace) command after the LV SYSTEM: prompt provides a means of defining the LV interface and status. A valid interface name is an interface name assigned to the LV multichannel interface by the VAX/VMS operating system, as shown in Figure 3-11. If an error condition occurs during setup, the interface status is turned off; otherwise a status of ON is used.

LV INTERFACE	
INTERFACE NAME	: WTA1:
	WTA1:
INTERFACE NAME	: WTA1:
INTERFACE STATUS:	ON

VALID ENTRIES
VALID INTERFACE NAME IS A SERIAL PORT ASSIGNED TO THE LV SYSTEM. VALID STATUS IS ON OR OFF.

Figure 3-11. Entering the LV Interface Name and Status

SC. Prior to testing, the TSI MI-990 multichannel LV interface must know the number of channels to process. This option allows the operator to specify the number of TSI 1990C LV counter processors connected to the MI-990. As shown in Figure 3-12, entering a valid channel number between one and three controls the number of channels. The software then forms and sends the MI-990 command to set the number of channels. If an error occurs in the transmission, then message is sent to the screen to notify the operator.

SET NUMBER OF LV COUNTER PROCESSORS			
ENTER NUMBER OF LV COUNTERS: 2			
: 2			
ENTER NUMBER OF LV COUNTERS: 2			
: █			
	<table border="1"><thead><tr><th>VALID ENTRIES</th></tr></thead><tbody><tr><td>VALID RESPONSE FOR NUMBER OF COUNTER PROCESSORS IS AN INTEGER NUMBER BETWEEN 1 AND 3</td></tr></tbody></table>	VALID ENTRIES	VALID RESPONSE FOR NUMBER OF COUNTER PROCESSORS IS AN INTEGER NUMBER BETWEEN 1 AND 3
VALID ENTRIES			
VALID RESPONSE FOR NUMBER OF COUNTER PROCESSORS IS AN INTEGER NUMBER BETWEEN 1 AND 3			

Figure 3-12. Entry of the SC (Set number of Channels) Command

SM. The SM (Set MI-990 interface) command from Figure 3-12 provides the means for tailoring the multichannel interface configuration for a particular run. Figure 3-13 lists the options available for controlling the interface. These include CO, CW, ET, EX, IT, RS, and TB. After entering the appropriate setup command, an instruction for carrying out the specified operation is formed and sent to the MI-990.

TSI MI-990 SETUP									
SET MI : █									
	<table border="1"><thead><tr><th>MI-990 SETUP COMMANDS</th></tr></thead><tbody><tr><td>CO - Set coincidence mode</td></tr><tr><td>CW - Prepare for C-words</td></tr><tr><td>ET - Sample at equal times</td></tr><tr><td>EX - Exit MI-990 setup</td></tr><tr><td>IT - Inhibit sampling time</td></tr><tr><td>RS - Randomly sample data</td></tr><tr><td>TB - Send time between data</td></tr></tbody></table>	MI-990 SETUP COMMANDS	CO - Set coincidence mode	CW - Prepare for C-words	ET - Sample at equal times	EX - Exit MI-990 setup	IT - Inhibit sampling time	RS - Randomly sample data	TB - Send time between data
MI-990 SETUP COMMANDS									
CO - Set coincidence mode									
CW - Prepare for C-words									
ET - Sample at equal times									
EX - Exit MI-990 setup									
IT - Inhibit sampling time									
RS - Randomly sample data									
TB - Send time between data									

Figure 3-13. Configuring the MI-990 interface for a given test

CO. In many applications, such as shear stress computation, all LV channels must contain valid data in order to be accepted. This coincidence must occur within a specified time interval controlled by the operator. The CO (COincidence) command shown in Figure 3-14 allows the operator to set this condition for the MI-990.

SET COINCIDENCE MODE	
CURRENT COINCIDENCE STATUS	: ON
	:
TIME DELAY INTERVAL (USEC)	: 5E1
	: 2E1
TIME DELAY INTERVAL (USEC)	: 2E1
	: █

VALID ENTRIES
VALID STATUSES - ON OR OFF
VALID TIMES ARE BETWEEN ONE
MICROSECOND AND ONE SECOND
FOR THE TIME DELAY

Figure 3-14. Entry of the TE command

CW. In addition to fringe crossing times and the time between samples, the MI-990 allows up to 16 additional inputs from various devices. These inputs are call "C word" inputs, with data from the LV counter processors forming the A and B words. The CW (C Word) command entry causes a screen display shown in Figure 3-15 to appear. This menu allows the MI-990 to acquire up to 16 channels of C word data or none at all.

SET MODE FOR C-WORD TRANSFERS	
CURRENT C-WORD MODE STATUS :	ON
:	:
ENTER NUMBER OF C-WORDS :	1
:	2
ENTER NUMBER OF C-WORDS :	2
:	1
ENTER NUMBER OF C-WORDS :	1
:	█
VALID ENTRIES	
VALID STATUSES - ON OR OFF	
VALID RESPONSES FOR NUMBER	
OF C-WORDS ARE BETWEEN 1	
AND 16 CHANNELS	

Figure 3-15. Specifying the number of C words

ET. Figure 3-16 shows the results of the ET command. Normally, the LV acquires data whenever a particle enters the measurement volume, which tends to occur at nonuniform time intervals. The ET (Equal Time interval) command forces sampling at uniformly spaced time intervals between one microsecond and one second. Of course, the time between particle arrivals should be much greater than the sampling interval.

SET EQUAL TIME SAMPLING (ETS) MODE	
CURRENT ETS ON/OFF STATUS	: OF
	: ON
CURRENT ETS ON/OFF STATUS	: ON
	:
TIME DELAY INTERVAL (USEC)	: 1E3
	: █

VALID ENTRIES
VALID STATUSES - ON OR OFF
VALID TIMES ARE BETWEEN ONE
MICROSECOND AND ONE SECOND
FOR THE TIME DELAY

Figure 3-16. Setting the Equal Time sampling mode in response to the ET command

IT. The IT command Inhibits the sending of the Time between samples by the MI-990 interface. As shown in Figure 3-17 entry of a status of on or off results in the time between samples being sent or inhibited. If the time between samples are inhibited, the time between data and even time sampling statuses are flagged as off.

INHIBIT SENDING TIME BETWEEN SAMPLES			
TIME BETWEEN SAMPLE STATUS : ON :			
BE SURE THAT ETS OR TBD IS ON			
ENTER <CR> TO CONTINUE			
	<table border="1"><thead><tr><th>VALID ENTRIES</th></tr></thead><tbody><tr><td>VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN TBD AND ETS STATUSES ARE SET TO OFF</td></tr></tbody></table>	VALID ENTRIES	VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN TBD AND ETS STATUSES ARE SET TO OFF
VALID ENTRIES			
VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN TBD AND ETS STATUSES ARE SET TO OFF			

Figure 3-17. Inhibiting the time between samples

RS. And for those times when demanding coincidence between samples simply will not do, try the RS (Randomly Sample) command. Here, every valid velocity sample triggers a transfer regardless of coincidence between samples. The result of entering RS are shown in Figure 3-18.

RANDOMLY SAMPLE LV DATA			
TIME BETWEEN SAMPLE STATUS : OF : BE SURE TO SET COINCIDENCE MODE ENTER <CR> TO CONTINUE			
	<table border="1"><thead><tr><th>VALID ENTRIES</th></tr></thead><tbody><tr><td>VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN THE COINCIDENCE MODE STATUS IS SET TO OFF</td></tr></tbody></table>	VALID ENTRIES	VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN THE COINCIDENCE MODE STATUS IS SET TO OFF
VALID ENTRIES			
VALID STATUSES - ON OR OFF IF OFF STATUS IS SELECTED THEN THE COINCIDENCE MODE STATUS IS SET TO OFF			

Figure 3-18. The results of entering the RS (Randomly Sampling) command

TB. The LV command TB (Time Between samples) signals the MI-990 to transfer time between samples along with the fringe crossing times. This additional information allows bias correction, conditional sampling, time series analysis of unsteady velocity data in addition to standard velocity statistics. Figure 3-19 shows the screen after the entry of this command.

SEND TIME BETWEEN SAMPLES	
TIME BETWEEN SAMPLE STATUS :	ON
:	:
TIME BETWEEN SAMPLE STATUS :	ON
:	■

VALID ENTRIES
VALID STATUSES - ON OR OFF IF ON STATUS IS SELECTED THEN SAMPLING INTERVALS ARE SENT WITH THE LV DATA

Figure 3-19. Screen displayed after entering the TB (Time Between samples) command

SR. For mapping flows about rotating machinery, TSI provides the Rotating Machinery Resolver. This software supports the RMR 1989 model. Entering the SR (Set RMR) command results in the display shown in Figure 3-20. The AD, IF, LD, MC, RM, and TC commands provided apply exclusively to the RMR and allow the instrument to be configured to provide the angular position through the C word of the MI-990, set with the SC command.

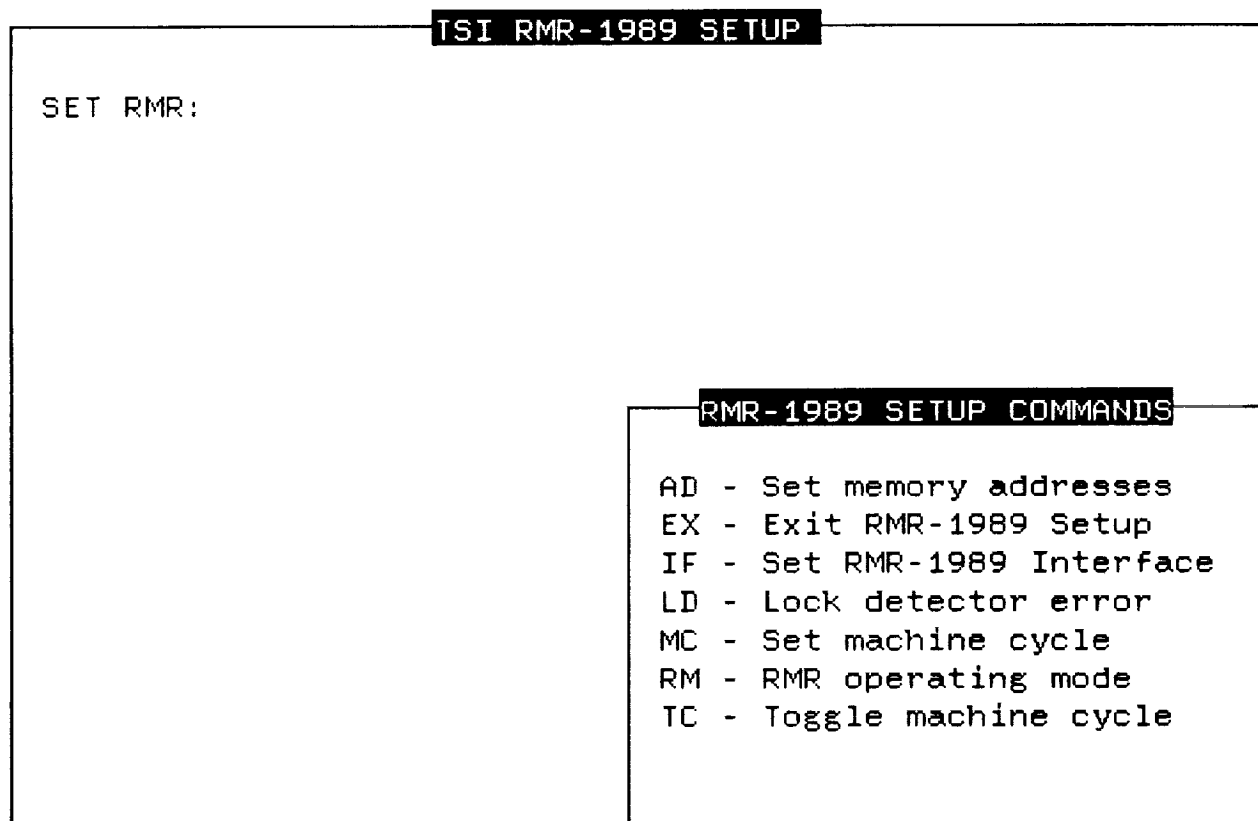


Figure 3-20. Screen display after the SR (Set RMR) command entry

AD. The RMR manual explains the procedure for setting up the RMR 1989 to acquire velocity data at specified angular intervals. This procedure consists of setting individual memory addresses from zero to 8191 either to zero or one. Each memory location corresponds to a one-tenth of a degree interval of shaft rotation. The command AD (set Address) controls the angular intervals when data is sampled or ignored by setting the RAM bit to 0 or 1.

SET RMR WINDOW PARTITION	
SET MEMORY SEGMENT ON/OFF:	1
	: 1
ENTER STARTING ADDRESS :	0
	:
ENTER STOPPING ADDRESS :	0
	: 4095
SET MEMORY SEGMENT ON/OFF:	1
	: <input type="checkbox"/>

VALID ENTRIES
Valid entries are
RAM bit = 0 ignores data
= 1 acquires data
Start/Stop Address =
0-4095 for 360 degrees
0-8191 for 720 degrees

Figure 3-21. Setting the conditional sampling intervals with AD (set Address) command

IF. Before any commands, valid or otherwise, can be sent to the RMR 1989, its interface must be defined. The IF (InterFace) command after the SR command provides this capability, as shown in Figure 3-22.

RMR INTERFACE	
INTERFACE NAME :	WTA2:
	WTA1:
INTERFACE NAME :	WTA1:
INTERFACE STATUS:	OFF
	ON
INTERFACE STATUS:	ON
	■

VALID ENTRIES
VALID INTERFACE NAME IS A SERIAL PORT ASSIGNED TO THE RMR-1989 SYSTEM. VALID STATUS IS ON OR OFF.

Figure 3-22. Setting the RMR interface with the IF (InterFace) command

LD. Not only can the sampling intervals be controlled, but the RMR 1989 allows the error tolerances in angular resolution to be controlled using the LD (Lock Detector sensitivity) command. As shown in Figure 3-23, entering an integer between 0 and 3 designates a specific tolerance.

SET RMR ERROR TOLERANCE	
LOCK DETECTOR SENSITIVITY :	0
	:
	█
VALID ENTRIES	
Valid entries are	
0, 1, 2, 3 for tolerances of	
96, 192, 384, 768 minutes	
respectively.	

Figure 3-23. The LD (Lock Detector) command specifies the angular sensitivity

MC. The command MC (Machine Cycle) sets the number of degrees per cycle of the RMR 1989 rotating machinery resolver. The setting allows either 360 or 720 degrees per cycle, as shown in Figure 3-24.

SET RMR MACHINE CYCLE	
MACHINE CYCLE SETTING : 1 :	
VALID ENTRIES	
Valid entries are 0 for 720 degrees/cycle 1 for 360 degrees/cycle	

Figure 3-24. The display in response to the MC (Machine Cycle) command

RM. Setting the RMR operating mode is the function of the RM (RMR Mode) command, which results in a display shown in Figure 3-25. The most common setting is PLL, phased-locked loop. But other settings, described in the RMR 1989 instrumentation manual, are available for double and quadruple pulse intervals as well as a micro mode.

SET RMR OPERATING MODE	
OPERATING MODE SELECTION : 0 :	
<div><div>VALID ENTRIES</div><div>Valid entries are 0 for phase-locked loop 1 for shaft encoder X2 2 for shaft encoder X4 3 for micro mode</div></div>	

Figure 3-25. Screen display for the RM (RMR Mode) command

TC. For those tests where phasing makes a difference, the TC (Toggle machine Cycle) allows the RMR cycle to be delayed one-half of a cycle and results in the display shown in Figure 3-26.

TOGGLE RMR MACHINE CYCLE	
TOGGLE MACHINE CYCLE	: Y
	: Y
TOGGLE MACHINE CYCLE	: Y
	: <input type="checkbox"/>

TOGGLE DESCRIPTION
Toggling the machine cycle to make sure the correct half cycle order occurs. Enter N OR <CR> to exit.

Figure 3-26. Display for Toggling the machine Cycle (TC command)

3.3.1.3 Controlling The Velocity Statistics

To set the variables controlling the LV velocity statistics, enter SS after the LV: prompt on the LV SETUP menu (or enter SE/LV/SS from the executive menu after the READY: prompt) as shown in Figure 3-27.

EXECUTIVE								
READY: SE/LV/SS READY: SE/LV/SS								
<table border="1"><thead><tr><th>EXECUTIVE COMMANDS</th></tr></thead><tbody><tr><td>AN - ANALYZE ACQUIRED DATA</td></tr><tr><td>EN - END AND RETURN TO VMS</td></tr><tr><td>RU - RUN</td></tr><tr><td>RR - RUN AND REDUCE DATA</td></tr><tr><td>SE - SETUP RUN VARIABLES</td></tr><tr><td>ST - SIMULATE A TEST RUN</td></tr></tbody></table>		EXECUTIVE COMMANDS	AN - ANALYZE ACQUIRED DATA	EN - END AND RETURN TO VMS	RU - RUN	RR - RUN AND REDUCE DATA	SE - SETUP RUN VARIABLES	ST - SIMULATE A TEST RUN
EXECUTIVE COMMANDS								
AN - ANALYZE ACQUIRED DATA								
EN - END AND RETURN TO VMS								
RU - RUN								
RR - RUN AND REDUCE DATA								
SE - SETUP RUN VARIABLES								
ST - SIMULATE A TEST RUN								

Figure 3-27. Entering SS to Control the Velocity Statistics

As shown in Figure 3-28a and 3-28b the SS command allows you to set the number of samples, confidence limit for histogram clipping (normally 98 per cent), histogram spacing (normally 1), and the option to turn histogram online displays on or off. This last option is currently not supported but included for future upgrades.

LV STATISTICS CONTROL	
NUMBER OF SAMPLES	1000
	500
NUMBER OF SAMPLES	500
CONFIDENCE LIMIT	98.0
	99.
CONFIDENCE LIMIT	99.0
	█

VALID ENTRIES
NUMBER OF SAMPLES ARE INTEGERS < 1,000,000
HISTOGRAM CONFIDENCE LIMITS ARE REAL NUMBERS > 1.0%
HISTOGRAM SPACING IS AN INTEGER BETWEEN 1 AND 100
HISTOGRAM OPTIONS ARE - ON TO VIEW ONLINE HISTOGRAMS OF - FOR NO VIEWING

Figure 3-28a. Setting the Number of Samples and Confidence Limit

LV STATISTICS CONTROL	
HISTOGRAM SPACING	10
	1
HISTOGRAM SPACING	1
HISTOGRAM OPTION	OF
	ON
HISTOGRAM OPTION	ON

VALID ENTRIES
NUMBER OF SAMPLES ARE INTEGERS < 1,000,000 HISTOGRAM CONFIDENCE LIMITS ARE REAL NUMBERS > 1.0% HISTOGRAM SPACING IS AN INTEGER BETWEEN 1 AND 100 HISTOGRAM OPTIONS ARE - ON TO VIEW ONLINE HISTOGRAMS OF - FOR NO VIEWING

Figure 3-28b. Setting the Histogram Spacing and Option

3.3.2 Test

TE. To enter the parameters describing the test, enter the command TE from the SETUP menu (or SE/TE from the executive menu). Figures 3-29a to 3-29c show the resulting screen display. Entries include the test number, tunnel, run number, code number, run title, angle of attack, length-to-chord ratio, reference velocity, and reference distance. Entering a carriage return or blanks retains the current value.

TEST DESCRIPTION	
TEST NUMBER	3
TEST TUNNEL	CE17
RUN NUMBER	1
CODE NUMBER	1

VALID ENTRIES
FACILITY - CE17, SE1, W7, UNK
TEST, RUN, CODE - MUST BE 0 < INTEGER < 1000
TITLE - UP TO 60 CHARACTERS
ALL OTHER ENTRIES ARE REAL NUMBERS

Figure 3-29a. Entering the Test Number, Facility, Run and Code

TEST DESCRIPTION			
TITLE Lewis Checkout			
REFERENCE DISTANCES	1.000	1.000	1.000
REFERENCE VELOCITY	100.00	100.00	100.00
REFERENCE ANGLES	0.00	0.00	0.00

VALID ENTRIES
FACILITY - CE17, SE1, W7, UNK TEST, RUN, CODE - MUST BE 0 < INTEGER < 1000 TITLE - UP TO 60 CHARACTERS ALL OTHER ENTRIES ARE REAL NUMBERS

Figure 3-29b. Entries for the test title, and reference distances, velocities, and angles

3.3.3 Traverse

TR. To control the positioning sequence during a run, utilize the TR (TRaverse) command while in the SETUP menu. Upon entry of this command, no response occurs, because the command is not currently supported. It has been included for completeness and future upgrades.

3.4 RUN

While in the executive menu, entering the RU (RUn) command after the READY: prompt results initiates the run sequence. The screen display during a run is shown in Figure 3-30. The information consists of test information (run and code), position (axis, coordinate), and velocity statistics (channel number, samples, means, and standard deviations). This capability is not currently supported but will be added after implementing the LVABI interface.

RUN	CODE	AXIS	POSITION	CHANNEL	SAMPLES		MEANS		STD	
1	1	X	0.000	1, 2	0,	0	0.0,	0.0	0.0,	0.0
		Y	0.000	3, 4	0,	0	0.0,	0.0	0.0,	0.0
		Z	0.000	5, 6	0,	0	0.0,	0.0	0.0,	0.0

Figure 3-30. Online information.

3.5 SIMULATE A TEST

Entering the command ST (Simulate a Test) at the executive menu display results in a run sequence using simulated data instead of actual velocity data from the TSI LV system. The velocity data are generated at an average sampling rate of 1000 samples per second with the time between samples distributed according to a Poisson ditribution. Currently, this capability supports software development. Future enhancements can include generation of simulated data allowing the operator to control the amplitude and frequency of the velocity fluctuations for checkout of advanced data reduction algorithms and intrument resolution.

4.0 ERRORS

The errors handled by this program are of two types - operator and instrument. Operator errors handled consist of erroneous commands and data entries. Erroneous commands result in the reissuance of the appropriate prompt with no action being taken. Erroneous data entries cause the current values of the variables to be retained.

Instrument errors result in message consisting of three parts. The first is the error source. The second part described the error condition. The third part gives a procedure for recovering from the error. An example of an error message is given in Figure 4-1 resulting from entering an invalid name for the LV MI-990 interface. This particular message results in the interface status being set to OFF to prevent possible problems in subsequent calls to the device.

LV INTERFACE				
INTERFACE NAME : WTA1: WTA2: ERROR IN LV INTERFACE *** NO SUCH DEVICE EXISTS *** --- SEE IF DEVICE INTERFACE EXISTS AND REENTER INTERFACE STATUS: OFF ■		<table border="1"><thead><tr><th>VALID ENTRIES</th></tr></thead><tbody><tr><td>VALID INTERFACE NAME IS A SERIAL PORT ASSIGNED TO THE LV SYSTEM. VALID STATUS IS ON OR OFF.</td></tr></tbody></table>	VALID ENTRIES	VALID INTERFACE NAME IS A SERIAL PORT ASSIGNED TO THE LV SYSTEM. VALID STATUS IS ON OR OFF.
VALID ENTRIES				
VALID INTERFACE NAME IS A SERIAL PORT ASSIGNED TO THE LV SYSTEM. VALID STATUS IS ON OR OFF.				

Figure 4-1. Example of an Error Message